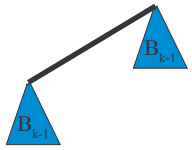
TD No5: tas binomiaux

Exercice 1 arbres binomiaux

Un arbre binomial B_k est soit l'arbre vide, soit construit à partir de deux arbres de la forme B_{k-1} en reliant la racine de B_{k-1} à celle de l'autre arbre de la forme B_{k-1} de la façon suivante:



Question 1 : exemples

Construisez B_0, B_1, B_2, B_3, B_4 et indiquez le nombre d'élément de chacun de ces arbres. Donnez un exemple d'arbre binomial pouvant contenir exactement 32 éléments. Faites de même avec un arbre binomial pouvant contenir exactement 7 éléments.

Question 2 Propriétés de B_k

Montrez que l'arbre binomial B_k vérifie les propriétés suivantes :

- il a 2^k éléments
- sa hauteur est k
- la racine a k fils et c'est le seul sommet qui vérifie cette propriété
- si l'on numérote les fils de la racine de droite à gauche, le premier est B_0, le second est B_1, le second est B_2, ..., le dernier est B_{k-1}.

Question 3 fusion de deux arbres binomiaux

On représente les arbres binomiaux sous la forme d'un arbre binaire fils/freres de la façon suivante:

```
typedef noeud * arbreBinomial ;
typedef struct tnoeud {
  arbreBinomial fils;
  arbreBinomial frere;
  arbreBinomial pere;
  int valeur;
  int degre;
} noeud;
```

Représentez graphiquement un arbre B_4 en vous appuyant sur cette définition.

Ecrire une fonction qui fusionne deux arbre binomiaux h et b de même taille. La racine de h deviendra la racine de l'arbre obtenu. Votre fonction modifierea sur place les champs de h et de b. Elle ne modifiera pas le père de h.

Question 4 suppression d'un noeud d'un arbre binomial

Ecrire une fonction ayant l'adresse d'un noeud d'un arbre binomial comme argument qui supprime

IUP2-Miage-ASR Structures de données 2004-2005

ce noeud de l'arbre auquel il appartient.

Exercice 2 Tas binomial

Un tas binomial est une forêt d'arbres binomiaux vérifiant les propriétés suivantes :

- La valeur de chaque noeud de chaque arbre binomial est inférieure à la valeur de ses fils.
- Les arbres binomiaux du tas sont tous d'ordre de taille différente
- les arbres binomiaux sont triés par taille croissante.

On souhaite pour réaliser efficacement les opérations suivantes sur nos tas :

- créer un tas vide
- insérer un élément dans un tas
- obtenir le minimum d'un tas
- obtenir et supprimer le minimum d'un tas
- réaliser l'union de deux tas
- diminuer la valeur d'un élément d'un tas
- supprimer un élément d'un tas

Question 1 : Exemples

Donnez des exemples de tas binomiaux à 8, 10 et 7 éléments. Montrez qu'un tas binomial a n éléments contient au plus $E(\log_2 n)$ arbres binomiaux.

Question 2 : implantation

On vous demande de proposer une représentation des tas binomiaux, de proposer les algorithmes permettant de réaliser les opérations suivantes et d'évaluer leur complexité:

- création d'un tas binomial vide
- obtention de la clef minimale
- union de deux tas
- ajout d'une clef
- suppression de la clef minimale
- diminution de la valeur d'une clef
- suppression d'une clef

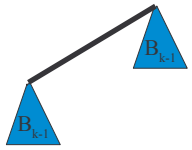
Question 3 : comparaison avec d'autres SD

Comparez la complexité de ces opérations avec celles sur les tas binaires vu dans le cadre du tri par tas.

TD No5: tas binomiaux

Exercice 1 arbres binomiaux

Un arbre binomial B_k est soit l'arbre vide, soit construit à partir de deux arbres de la forme B_{k-1} en reliant la racine de B_{k-1} à celle de l'autre arbre de la forme B_{k-1} de la façon suivante:



Question 1 : exemples

Construisez B_0, B_1, B_2, B_3, B_4 et indiquez le nombre d'élément de chacun de ces arbres. Donnez un exemple d'arbre binomial pouvant contenir exactement 32 éléments. Faites de même avec un arbre binomial pouvant contenir exactement 7 éléments.

Éléments de correction:

Un arbre binomial contient 2^k éléments. Il ne peut donc exister d'arbres à 7 éléments. :-)

Question 2 Propriétés de B_k

Montrez que l'arbre binomial B_k vérifie les propriétés suivantes :

- il a 2^k éléments
- sa hauteur est k
- la racine a k fils
- si l'on numérote les fils de la racine de droite à gauche, le premier est B_0, le second est B_1, le second est B_2, ..., le dernier est B_{k-1}.

Éléments de correction:

ces propriétés se démontrent facilement par récurrence en s'appuyant sur la définitions des arbres binomiaux.

Question 3 fusion de deux arbres binomiaux

On représente les arbres binomiaux sous la forme d'un arbre binaire fils/freres de la façon suivante:

```
typedef noeud * arbreBinomial ;
typedef struct tnoeud {
  arbreBinomial fils;
  arbreBinomial frere;
  arbreBinomial pere;
  int valeur;
  int degre;
} noeud;
```

Représentez graphiquement un arbre B_4 en vous appuyant sur cette définition.

Ecrire une fonction qui fusionne deux arbre binomiaux h et b de même taille. La racine de h deviendra la racine de l'arbre obtenu. Votre fonction modifierea sur place les champs de h et de b. Elle ne modifiera pas le père de h.

Question 4 suppression d'un noeud d'un arbre binomial

Ecrire une fonction ayant l'adresse d'un noeud d'un arbre binomial comme argument qui supprime ce noeud de l'arbre auquel il appartient.

Exercice 2 Tas binomial

Un tas binomial est une forêt d'arbres binomiaux vérifiant les propriétés suivantes :

- La valeur de chaque noeud de chaque arbre binomial est inférieure à la valeur de ses fils.
- Les arbres binomiaux du tas sont tous d'ordre de taille différente
- les arbres binomiaux sont triés par taille croissante.

On souhaite pour réaliser efficacement les opérations suivantes sur nos tas :

- créer un tas vide
- insérer un élément dans un tas
- obtenir le minimum d'un tas
- obtenir et supprimer le minimum d'un tas
- réaliser l'union de deux tas
- diminuer la valeur d'un élément d'un tas
- supprimer un élément d'un tas

Question 1 : Exemples

Donnez des exemples de tas binomiaux à 8, 10 et 7 éléments. Montrez qu'un tas binomial a n éléments contient au plus $E(\log_2 n)$ arbres binomiaux.

Éléments de correction:

On peut faire un parallèle entre les arbres binomiaux composant un tas et la décomposition en base 2 du nombre d'éléments du tas. Ainsi:

- *$8 = 2^3$ donc le tas à 8 éléments est constitué du seul arbre B_3*
- *$10 = 2^3 + 2^1 + 2^0$ donc le tas à 10 éléments est constitué des arbres $B_0;B_1;B_3;$*
- *$7 = 2^2 + 2^1 + 2^0$ donc le tas à 7 éléments est constitué des arbres $B_0;B_1;B_2;$*

Si $t= B_i, ..., B_{k-1}$. Le nombre d'éléments n de t est supérieur au nombre d'éléments de B_{k-1}. On en déduit que $2^{k-1} \leq n$ et donc que $\log 2^{k-1} \leq \log n$ et donc que $k-1 \leq \log n$.

Comme on travaille avec des entiers, on en déduit: $k < \log n$

Question 2 : implantation

On vous demande de proposer une représentation des tas binomiaux, de proposer les algorithmes permettant de réaliser les opérations suivantes et d'évaluer leur complexité:

- création d'un tas binomial vide
- obtention de la clef minimale

- union de deux tas
- ajout d'une clef
- suppression de la clef minimale
- diminution de la valeur d'une clef
- suppression d'une clef

éléments de correction:

```
cf leicerson.
```

Question 3 : comparaison avec d'autres SD

Comparez la complexité de ces opérations avec celles sur les tas binaires vu dans le cadre du tri par tas.